

AFOSR-TR- 90 - 0 2 3 7

TR/90-01

AD-A218 868

# Intelligent Real-Time Problem-Solving: Issues, Concepts and Research Methodology

January 25, 1990

Prepared by:  
Stanley J. Rosenschein  
TELEOS RESEARCH  
576 Middlefield Road Palo Alto, CA 94301

With contributions from:  
Michael Fehling (Stanford University)  
Matthew L. Ginsberg (Stanford University)  
Eric J. Horvitz (Stanford University)  
Bruce D. D'Ambrosio (Oregon State University)

Prepared for:  
Air Force Office of Scientific Research  
Bolling Air Force Base  
DC 20332-6448

Contract Number F49620-89-C-0117

DTIC  
ELECTE  
FEB 26 1990  
S B D

DISTRIBUTION STATEMENT A

Approved for public release;  
Distribution Unlimited

90 02 23 054

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.	
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TR/90-01		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION TELEOS RESEARCH	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFOSR	
6c. ADDRESS (City, State, and ZIP Code) 576 MIDDLEFIELD ROAD PALO ALTO, CA 94301		7b. ADDRESS (City, State, and ZIP Code) Bldg 410 Bolling, AFB, DC 20332-6448	
8a. NAME OF FUNDING / SPONSORING ORGANIZATION AFOSR	8b. OFFICE SYMBOL (If applicable) NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F49620-89-C-0117	
8c. ADDRESS (City, State, and ZIP Code) BUILDING 410 BOLLING AIR FORCE BASE WASHINGTON, D.C. 20332-6448		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO. 162702F	PROJECT NO. 5581
		TASK NO. A7	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) INTELLIGENT REAL-TIME PROBLEM-SOLVING: ISSUES, CONCEPTS AND RESEARCH METHODOLOGY (U)			
12. PERSONAL AUTHOR(S) STANLEY J. ROSENSCHEIN			
13a. TYPE OF REPORT CONTRACTOR REPORT: FINAL	13b. TIME COVERED FROM AUG 89 TO NOV 89	14. DATE OF REPORT (Year, Month, Day) 90/01/25	15. PAGE COUNT 37
16. SUPPLEMENTARY NOTATION STANLEY J. ROSENSCHEIN: TELEPHONE 415/328-8800 AFOSR PROGRAM MANAGER, DR. ABRAHAM WAKSMAN: 202/767-5025			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
		REAL-TIME, PROBLEM-SOLVING, KNOWLEDGE-BASED SYSTEMS, EMBEDDED SYSTEMS, INTELLIGENT AGENTS, Real-time Systems, Computer Systems	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>The Air Force has sponsored a study aimed at laying out research issues in the area of intelligent real-time problem-solving. As part of this study, a team led by Dr. Stanley J. Rosenschein of Teleos Research, has reviewed topics in this area and has participated in a workshop. This report contains a position statement of the Teleos team prepared for that workshop, along with a discussion of the research issues panel held at the workshop itself and of methodologies for evaluating intelligent real-time problem-solving systems.</p>			
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. A. Waksman		22b. TELEPHONE (Include Area Code) (202) 767-5028	22c. OFFICE SYMBOL NM

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Team Position Statement for IRTPS Workshop</b>	<b>5</b>
2.1	Background . . . . .	5
2.2	General Programmatic Issues . . . . .	6
2.3	Historical/Interdisciplinary . . . . .	7
2.4	Core Research on Resource-Bounded Reasoning . . . . .	7
2.4.1	Base-level/Meta-level Reasoning . . . . .	8
2.4.2	Anytime Reasoning . . . . .	8
2.4.3	Compilation (Compiled vs. Deliberative Reasoning) . . . . .	9
2.4.4	Pre-Emptive Control . . . . .	9
2.5	Experimental Validation . . . . .	9
<b>3</b>	<b>Notes on Methodologies for Evaluating IRTPS Systems</b>	<b>11</b>
3.1	Introduction . . . . .	11
3.2	A Model of Embedded Systems . . . . .	12
3.3	Measurement and Utility . . . . .	12
3.4	Evaluation . . . . .	14
3.4.1	Analytical Evaluation Methods . . . . .	14
3.4.2	Experimental Evaluation Methods . . . . .	15
3.5	Other Implications . . . . .	17
3.5.1	Implications for Requirements Specification . . . . .	18
3.6	Implications for an Experimental Testbed . . . . .	19
<b>4</b>	<b>Report on Research Issues Session of IRTPS Workshop</b>	<b>20</b>

4.1	Introduction . . . . .	20
4.2	Candidate IRTPS Paradigms . . . . .	22
4.2.1	Planning . . . . .	22
4.2.2	Logics . . . . .	22
4.2.3	Algebraic Techniques . . . . .	23
4.2.4	Decision Theory . . . . .	23
4.2.5	Control Theory . . . . .	23
4.2.6	State-Based Automata Methods . . . . .	24
4.2.7	No-Paradigm . . . . .	24
4.3	Toward an IRTPS Research Agenda . . . . .	25
4.3.1	Research Areas . . . . .	25
4.3.2	Prioritization . . . . .	25
4.4	Concluding Remarks . . . . .	28
5	Annotated Bibliography . . . . .	29



<b>Accession For</b>	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
<b>Availability Codes</b>	
Dist	Avail and/or Special
A-1	

4.1	Introduction . . . . .	20
4.2	Candidate IRTPS Paradigms . . . . .	22
4.2.1	Planning . . . . .	22
4.2.2	Logics . . . . .	22
4.2.3	Algebraic Techniques . . . . .	23
4.2.4	Decision Theory . . . . .	23
4.2.5	Control Theory . . . . .	23
4.2.6	State-Based Automata Methods . . . . .	24
4.2.7	No-Paradigm . . . . .	24
4.3	Toward an IRTPS Research Agenda . . . . .	25
4.3.1	Research Areas . . . . .	25
4.3.2	Prioritization . . . . .	25
4.4	Concluding Remarks . . . . .	28

# Chapter 1

## Introduction

Knowledge-based systems technology has led to practical applications in a number of areas, but to date it has not produced adequate techniques for dealing with systems that must interact intelligently with their environments in real time. To address this need, the Air Force has launched an initiative aimed at stimulating a national research effort on Intelligent Real-Time Problem Solving (IRTPS).

As part of that study, a team of five researchers, led by Dr. Stanley J. Rosenschein of Teleos Research with the assistance of Professor Michael Fehling of Stanford University, has investigated a variety of topics and has formulated positions on the research area. This work was augmented by discussions with members of other IRTPS teams and colleagues in the research community. This document contains the final report of the Teleos team.

Before describing the contents of the report, however, let us review the activities in which team members engaged as part of the IRTPS project.

- Exchange of background documents and comments on IRTPS.
- Discussions prior to program kick-off meeting.
- Participation in one-day kick-off meeting at Cimflex Teknowledge, Palo Alto.
- Several meetings between S. Rosenschein and M. Fehling to discuss research issues.
- Written contributions by all team members.
- Two pre-workshop meetings to discuss and finalize position paper.
- A series of meetings of S. Rosenschein with Lee Erman of Cimflex Teknowledge and Barbara Hayes-Roth of Stanford University to discuss research methodology and experimental evaluation methods.
- Written contributions by S. Rosenschein leading to a paper on methodologies for IRTPS research.
- Participation in IRTPS workshop in Santa Cruz.

- Panel on Research Issues chaired by Teleos team members.
- Written contributions summarizing discussion of Research Issues panel.
- Submission to Lee Erman of written Architecture documents and Annotated Bibliography.
- Preparation of Final Report.

This Final Report contains three documents. The first document (Chapter 2) is the Team Position Statement prepared by the Teleos team for the IRTPS workshop. It contains a discussion of terms and issues and highlights certain recommendations of a programmatic nature regarding the management of the national IRTPS effort. The second document (Chapter 3), co-authored by Dr. Stanley Rosenschein of Teleos, Dr. Barbara Hayes-Roth of Stanford, and Dr. Lee Erman of Cimflex Teknowledge, addresses issues of research methodology, especially the experimental evaluation of the performance of embedded real-time problem solving systems. It states desiderata and criteria that could influence the design of an experimental IRTPS testbed. The third document (Chapter 4), prepared collaboratively by Teleos team members, summarizes the discussion held as part of the Research Issues panel at the IRTPS workshop in Santa Cruz. These documents are followed by an IRTPS bibliography.

## Chapter 2

# Team Position Statement for IRTPS Workshop

Over the last decade and a half, advances in knowledge-based systems technology have led to practical applications in a variety of problem domains. Despite these advances, current technology remains inadequate for dealing with systems that must maintain real-time interactions with ongoing processes in their environment. Because systems of this type are critical to many important applications, particularly in defense, the Air Force has launched an initiative aimed at stimulating the development of a national research effort on Intelligent Real-Time Problems Solving (IRTPS).

The goal of the first phase of the research initiative is to clarify terms and issues underlying intelligent real-time problem solving, and as part of this effort three research groups, including the Teleos group led by Stanley Rosenschein, have been chosen to investigate these issues to help guide subsequent phases of the program. Phase 1 is to culminate in a workshop at which the groups compare their findings and discuss issues with invited researchers. This document contains an interim report of the Teleos group and is intended to serve as a draft position paper for the IRTPS Workshop. It contains a preliminary review of terms and issues followed by a discussion of implications for the IRTPS research program being undertaken by the Air Force.

### 2.1 Background

In the simplest terms, intelligent real-time problem-solving systems are characterized by the following features:

1. They take in a stream of sensory input from the environment.
2. They produce a stream of control output that affects the environment.
3. The intervening computation is modeled as a reasoning or problem-solving process.



#### 4. Time matters.

The primary challenge in developing systems of this type lies in reconciling the conflict inherent in the last two attributes. In conventional knowledge-based applications, the system is intended to provide support to human problems solvers, where the humans have sole responsibility for real-time interaction with the environment, and the system is not required to exhibit real-time performance. For example, these systems are typically unable to adapt their mode of operation to changing deadlines for an action or the unpredicted occurrence of critical events.

The central programming model for conventional knowledge-based systems work is that of an inference engine that performs reasoning steps and draws conclusions from a set of domain-specific rules or facts stored in a knowledge base. Because chains of inference leading to conclusions can vary greatly in length and draw on facts in the knowledge base in ways that are hard to predict and control, it is difficult to bound the execution time of programs in this model. This is compounded by the difficulty of controlling the performance of the operating systems environment (e.g., list-processing) in which most knowledge-based systems are embedded. In fact, much of the appeal of the knowledge-based model lies precisely in the fact that it abstracts away from the details of resource allocation required to support inference and allows the programmer to deal primarily with the content of the inferences. However, in real-time applications, the resource question cannot be ignored. Without considering execution time, rates of inference cannot be related to rates of change in the environment, and the designer cannot be sure that the system will be able to find satisfactory output responses in a timely fashion.

## 2.2 General Programmatic Issues

We see the IRTPS program as addressing the need for real-time knowledge-based systems by developing three mutually-supportive research thrusts:

1. Historical/Interdisciplinary
2. Core Research on Resource-Bounded Reasoning
3. Experimental Validation

The allocation of funds among and within these thrusts should be at least partially "bottom-up," i.e., it should be responsive to the best ideas offered during proposal solicitation. However, an effort should be made to maintain balance so that major areas are not entirely neglected.

Because of the limited funding allocated to the main research effort of the IRTPS program (\$800,000 over an 18-month period), a realistic objective for the program is to seed research in each of the key areas and to establish research paradigms and activities to which additional funding can be attracted. This is especially true for the experimental component of the

program; the resources required to develop and distribute a realistic testbed, for instance, could overwhelm the program's funding, leaving little for basic research unless special care is taken to leverage existing software and other government programs.

In the following sections, we describe each of the proposed program thrusts and list several research questions that might be explored in each. These questions are meant to be suggestive rather than exhaustive, and will undoubtedly be supplemented as the research proceeds.

## **2.3 Historical/Interdisciplinary**

The current IRTPS program does not exist in a vacuum. IRTPS systems have been built using existing artificial-intelligence (AI) concepts and tools, and these should be investigated with a view toward drawing out the lessons to be learned. Among the approaches that might be taken to these investigations are case studies, literature searches, and attempts to classify existing systems in terms of the conceptual categories developed in the IRTPS program.

In addition, artificial intelligence is not the only discipline to be concerned with embedded real-time computation or intelligent problem solving. Work in decision theory and control theory, in real-time operating systems and scheduling algorithms, and in computer-based control systems has resulted in a large body of practice, theory, and engineering methodology. Research in these disciplines should be explored in light of IRTPS requirements and objectives. In particular, methods should be developed that will allow non-AI approaches to be adapted and applied to IRTPS problems and to coexist with specialized techniques developed in the AI framework.

## **2.4 Core Research on Resource-Bounded Reasoning**

This thrust should be aimed at developing a deeper understanding of the tradeoffs inherent in reasoning under time stress. Constraints on a real-time reasoning system's inference and representation lead to inescapable uncertainties about the problems that may be faced. A real-time system immersed in a complex world must grapple with uncertainty associated with both the environment (object-level) and the reasoner (inference level). In addition, agents must typically contend with deep uncertainty about the value of future reasoning. However the benefits of controlling computational tradeoffs in theoretically coherent ways are very great in high-stakes decision-making arenas such as medicine, aerospace, and defense, and justify an intensive research effort in this area. An important part of this research will involve a synthesis of logical models of reasoning with Bayesian and, more generally, decision-theoretic models.

### 2.4.1 Base-level/Meta-level Reasoning

One set of research issues focuses on the control of search as a method for bounding execution time of reasoning processes. Reasoning is modeled as a search process in which many potential branches are available for exploration and in which choices are made about where effort should be allocated. The computation is broken into elementary bounded-time steps (with different approaches varying in the grain size they consider for these elementary units), and attention is shifted under the control of a higher-level process whose time behavior is well understood. This method of resource allocation similar to that used in a multi-tasking operating systems, and, as in the case of operating systems, interruptability, pre-emption, and prioritization are the key terms of analysis. Unlike operating systems, however, issues of the *content* of the computation as a reasoning process need to be more fully modeled and related to the resource-allocation algorithm.

Considerations of resource allocation lead directly to questions regarding the *criteria* by which allocation is to be judged. One approach (metalevel control of reasoning) is to formulate explicit theories about the reasoning process and the effect of alternative control strategies. Research is needed on methods for allocating resources between base-level and meta-level reasoners. An important goal of this research is to discover ways of limiting the amount of time spent doing meta-level reasoning, or more generally, to optimize the split of resources between base- and meta-level reasoning. An attempt should be made to identify classes of tractable, closed-form, metalevel control problems.

For many applications it will be important to quantify (1) the value or cost of achieving (or not achieving) goals, (2) uncertainty about the existence of alternative states of the world, and (3) the costs of continuing to deliberate (versus taking an action). Decision theory provides a useful framework for the design and evaluation of real-time systems as it gives us a language and precise semantics for capturing preferences and uncertainty.

### 2.4.2 Anytime Reasoning

Algorithms that can produce partial or reduced-quality output when their execution is terminated before some complete solution is produced have been called *anytime algorithms*. Useful properties that such algorithms can exhibit include monotonicity (improvement over time), continuity (gradual improvement), and convergence in the limit. We see a need to extend this line of research to include inference processes, so that so that declarative computations can be interrupted before they have finished running and still produce useful results.

Among the approaches that can be tried are the following:

1. Come up with an anytime inference algorithm that gradually and uniformly approaches the right answer.
2. Come up with an anytime inference algorithm that approaches the right answer in the large-runtime limit, but might wander around before doing so. (Presumably, the "quality" of the answer would increase uniformly, in some strange sense.)

If the second approach becomes necessary, one important question will be how to adapt the reasoning process, and the strategies that guide it, to changes in the environment that invalidate previous input or modify the available problem solving resources. Truth-maintenance techniques will be important here, though research will be required to adapt these techniques to the real-time setting. A related set of research topics involve modeling anytime probabilistic and decision-theoretic reasoning.

### 2.4.3 Compilation (Compiled vs. Deliberative Reasoning)

So far, we have only discussed deliberative approaches to reasoning and metareasoning. It can be important to reduce complex deliberation in computer-based reasoners by developing decision-making techniques that rely to some extent on precomputed or *compiled* responses. Such knowledge can be generated at design time or learned by agents over their lifetimes.

Recent research, including that labeled *reactive planning*, has centered on the replacement of unwieldy solution mechanisms and detailed representations of knowledge with compiled *situation-action* rules. Such rules enable agents to respond immediately perceptual inputs. Investigators have sensed that, for many contexts, explicit representations and deliberation will not be necessary for good performance.

Deliberation and reaction are merely two ends of a spectrum with many intermediate points. One important topic of research involves developing methods for optimizing the split between deliberation and reaction in the design of embedded agents.

### 2.4.4 Pre-Emptive Control

This sub-area should explore pre-emptive control strategies for multiple, simultaneous problem-solving activities, with a distinction being drawn between pre-emption and multitask management. For example, pre-emption can occur in managing even a single line of control. The policies that guide pre-emption, and the systems architecture that facilitate the implementation of these policies, need to be characterized and studied.

## 2.5 Experimental Validation

While abstract conceptual models of real-time reasoning are an important first step toward the design of practical IRTPS applications, these models must be augmented by a software-development methodology that designers can actually use to solve real-world problems. The methodology should help the programmer instantiate the general model to the particular data structures and operations required to satisfy the requirements of his particular application problem. The methodology includes specialized software-development tools that capture key abstractions, hide implementation details, but leave the programmer with sufficient flexibility and control over what is important. One goal of the IRTPS research program should be to enumerate and taxonomize programming methodologies, architectures, and

tools, relating them to the underlying conceptual model they support, and exploring whether the abstractions they provide can be made orthogonal and be incorporated into a more embracing IRTPS software methodology.

One of the goals of the IRTPS program is to provide a methodology for exploring architectures for real-time problem solving in an empirical setting. One way of promoting this goal is by establishing a common conceptual framework to guide experimental work in the IRTPS community. Program management has established an Experimental Methodologies Working Group which has outlined such a framework and produced a draft document describing its findings. The document describes ways that models of intelligent embedded systems might be modeled and evaluated, focusing specifically on comparing the effectiveness of agents with different compositions and abilities immersed in distinct problem contexts. Evaluation methods include theoretical analysis as well as experimental methods, both in simulated and real environments. The document goes on to describe what kind of controls would be necessary to make the results of experimental methods empirically meaningful and proposes a variety of measurement types relevant to real-time problem solving. Our team has reviewed a preliminary draft of this document and, in general terms, is in agreement with its analysis and recommendations.

A second, more concrete, way of enhancing the field's experimental methodology is to provide a common testbed that might be used by the IRTPS community to carry out empirical investigations. Such a testbed would allow precise measurements of the performance of systems and architectures. We feel such a testbed would be a useful component of the IRTPS program, provided it can be provided at reasonable cost and can be configured to adhere to the methodological guidelines proposed in the Experimental Methodologies Working Group document. One method of leveraging the program's research funds to good advantage would be to identify existing testbeds produced under other government programs that could be adapted to support experimental work in IRTPS. Consideration should be given to dimensions of variability among application domains, for example discrete vs. continuous domains, low-level vs. high-level perceptual data, and so on.

## Chapter 3

# Notes on Methodologies for Evaluating IRTPS Systems

*With Dr. Barbara Hayes-Roth (Stanford University) and Dr. Lee Erman (Cimflex Teknowledge)*

We propose a framework for modeling intelligent real-time problem-solving systems embedded in an environment. Within this framework, measurements may be defined on the system and on the environment, and particular measurements may be designated for judging the performance of the system. Although this framework supports analytical evaluation, we concentrate on its use for experimental evaluation, especially for evaluating and comparing system architectures. This framework also provides a basis for formalizing various requirements terms, such as "reactivity" and "graceful degradation".

### 3.1 Introduction

Intelligent real-time problem-solving systems are embedded computer systems that interact with their environments in a continuous fashion, sensing asynchronous events and acting in ways designed to satisfy certain goals. Instances of such systems include intelligent robots, factory control systems, avionic systems, and medical monitoring systems. Many software architectures have been proposed to ease the design and implementation of effective IRTPSs, and there has arisen the need for some objective means of evaluating and comparing them. As part of the research program on Intelligent Real-Time Problem Solving being sponsored by the Air Force, a small working group was formed to consider the methodologies for experimentally evaluating IRTPS architectures. This document is a preliminary report of that working group.

## 3.2 A Model of Embedded Systems

The first step in developing an evaluation methodology for IRTPSs is to lay out a conceptual framework for modeling systems and their interactions with the environment.

A natural beginning point is with models of dynamic physical systems. Such systems can be described in terms of time series of physical states, for example as mappings (possibly stochastic) from instants of time to some state space of values that model the physical features of interest. One then partitions the overall physical system into sub-components corresponding to the *IRTPS*  $S$  and the *environment*  $E$ , each having dynamic local state that varies as a function of signals received from the other. We may wish to regard the IRTPS and its environment as being parametrized in various ways. Let  $S(u)$  and  $E(v)$  represent the system and its environment with parameters  $u$  and  $v$  respectively. In addition, either or both of  $S$  and  $E$  could have random elements.

Because  $S$  and  $E$  are dynamic, time-varying objects, we are usually interested in describing not only those properties that hold statically of individual states, but properties of the time series of states. For present purposes, we will call these time series *runs* and write  $runs((S, E))$  to represent the set of runs of the combined IRTPS/environment pair. Each *run* is (conceptually) a sequence of total system states (i.e., states of the IRTPS/environment aggregate.) Conceptually, a run could be infinite and there could be an infinite number of runs.

Under this very general model, the boundary between  $S$  and  $E$  is arbitrary and can be adjusted to address different goals. In the present context, we are concerned with evaluating proposed IRTPS architectures with respect to their performance in particular classes of environments. Thus, the  $S - E$  boundary should be placed so as to distinguish between a proposed architecture and the environment in which it is claimed to be effective. Then we can evaluate the relationships between properties of the architecture as manifested in  $S$  and properties of  $E$ . In particular, note that the  $S - E$  boundary need not correspond to the boundary between a "complete agent" and its environment, but may correspond to the boundary around any "partial agent" of interest. For example, to evaluate a complete agent architecture, the  $S - E$  boundary should encompass all perception, reasoning, and action elements. But to evaluate a perception architecture, the  $S - E$  boundary should more tightly encompass only perceptual elements, with any reasoning or action elements treated as part of the environment. We might often choose to treat particular sensors and effectors as elements of  $E$ . As discussed below, for a given placement of the  $S - E$  boundary, we will be attempting to attribute properties in the environment to the behavior of particular IRTPSs and, by inference, to their underlying architectures.

## 3.3 Measurement and Utility

In order to describe the effectiveness of an IRTPS architecture in controlling aspects of the environment, it is necessary to identify measurements that can be made and how those measurements will be interpreted to determine utility.

A measurement is any function of state values. Measurements can be made within a state or over sets of states or runs.

Under the above model of embedded systems, for a given  $S - E$  boundary, measurements on  $E$  are distinguished from measurements on  $S$ . Measurements on  $E$  describe the dynamic properties of the environment, some of which are determined by processes internal to the environment and others of which are influenced by the behavior of  $S$  in  $E$ . The latter sorts of measurements are distinguished and used to assess the effectiveness of  $S$  in determining properties of  $E$  under various conditions. These assessments may be in absolute terms or relative to alternative  $S$ s. Measurements on  $S$  describe the dynamic properties of the IRTPS, some of which are determined by processes internal to it and others of which are determined by the impact of  $E$  on  $S$ . These measurements are used help to explain the performance of  $S$  and its (in)effectiveness in determining properties of  $E$  in terms of its underlying architecture. They also are used to analyze and predict its performance under other values of  $u$  and  $v$ .

Two classes of measurements are distinguished - descriptive and utility. Descriptive measurements represent objective features of a state, run, or set of runs. Examples are: (a) deadline  $d$  was met; and (b) 80% of deadlines of type  $t$  were met. Other illustrative simple descriptive measurements are:

- Latency from environmental event  $e_1$  to environmental event  $e_2$  e.g., latency from occurrence of a fault to occurrence of its correction
- Deadline satisfaction e.g., whether a given fault is corrected by the time of its deadline
- Logical correctness of result
- Quality of result
- Precision of result

Illustrative functions on measurements are:

- Average latency for critical events
- Percent deadline satisfactions for critical events
- Sum of latencies for all instances of event-type-a to event-type-b
- Average percentage over deadline on soft-deadline events

Utility measurements represent valuational conclusions based on the features or qualities of a state, run, or set of runs. An example is: satisfactory performance requires meeting  $> 95\%$  of priority 1 deadlines and  $> 50\%$  of priority 2 deadlines. Other illustrative utility measurement (higher is better) are:

- Weighted sum of importance  $\times$  deadline satisfied (0 or 1) for all events



- Weighted average of response quality  $\times$  importance  $\times$  deadline satisfied
- Gracefulness of degradation (suitably formalized)

The choice of utility measures may be specific to the  $S - E$  boundary placement. They certainly will be specific to the purpose of the evaluation.

To describe system  $S_1(u)$  parametrically with respect to various measurements of utility against fixed (parametric) environment  $E(v)$  amounts to characterizing the expected utility of  $S_1(u)$  as a function of  $\langle u, v \rangle$ , using a variety of techniques, some of which are described below. Similar methods can be used to compare two similarly parametrized systems,  $S_1(u)$  and  $S_2(u)$ , fixed (parametric) environment  $E(v)$ .

## 3.4 Evaluation

In principle, there are many ways a proposed IRTPS design might be evaluated. The methods fall into two general categories: *analytical* and *experimental*. Because each of these methods has its advantages and drawbacks, a thorough evaluation often requires both. In the first section below, we briefly mention some of the advantages and disadvantages of analytical methods. The following section treats experimental methods, which are the focus of this document, in more detail.

### 3.4.1 Analytical Evaluation Methods

One method of characterizing system performance is by establishing certain of its properties through analytical techniques. These techniques draw on relevant mathematical methods and amount to proving theorems.

The main advantage of the analytical approach is the ability to establish with mathematical certainty very general or universal statements about entire classes of behaviors and phenomena far too numerous to enumerate in explicit detail. For instance, it might be shown mathematically that a certain undesired situation can *never* arise given the nature of the environment and the control system, or that when a triggering event occurs a response is *always* generated within a certain time period. Results of this kind can be very powerful and can give us great confidence in the systems we design.

Analytical approaches are not without their drawbacks, however. The primary drawback is that the complexity of the systems being modeled gives rise to intractable mathematical models that often resist analysis. In an attempt to render the models tractable, simplifications are often made which cause the models to diverge from reality in ways that undercut the usefulness for building the model in the first place.

In defense of analytical methods it should be pointed out that there is no way to insure against bad modeling and analysis. Experience shows, however, that formal models are of use but often they must be supplemented by other techniques.

### 3.4.2 Experimental Evaluation Methods

The other main category of evaluation technique is experimental. In this approach, evaluation is done by measuring properties of particular instances of runs of particular systems and drawing certain conclusions. Experimental approaches can be further subclassified according to whether they involve (a) the real control system embedded in the real environment, (b) the real control system embedded in a simulated environment, (c) a simulated control system in a simulated environment, or (d) some hybrid approach. Real systems offer the obvious advantages of evaluating against reality, but they are often cumbersome or even unavailable, may pose unacceptable risks, etc. In the simulation approach to experimental evaluation, the environment and possibly the control system as well (e.g., if we are using a conventional machine to simulate a massively parallel system controlling an environment) a computer program is run to generate samples of the behavior of the overall system. These samples are then analyzed empirically to provide evidence in favor of certain conclusions regarding performance of the real system in the real environment.

Experiments provide a framework for inductive inference of general relations between architectures and real-time performance based on observations. The goal is to discover general relations that can be expected to hold whenever the appropriate conditions hold. For example, one relation might be that architecture *A* provides graceful degradation in performance under increasing rates of environmental events; another relation might be that architecture *B* produces unacceptable performance degradation under similar circumstances. Because it is infeasible to make observations of all of the instances encompassed by a hypothesized relation, it becomes necessary to draw conclusions about what would happen for all such instances based on observation of a few particular instances. For example, we might draw conclusions about the relative advantages of architectures *A* and *B* on the basis of their performance on a small number of environmental scenarios.

Drawing general conclusions from a small number of observed instances is a risky business. A given *S* realizes an abstract architecture, *A*, in a particular implementation, *I*, and instantiates it for a body of knowledge, *K*. Architectures themselves are complex artifacts, differing in both theoretically interesting variables (e.g., knowledge representation, inference procedures, control mechanism) and incidental variables (e.g., implementation details, execution environment). Similarly, different environmental scenarios differ in a great many variables (e.g., frequency of important events, distribution of deadlines, amount of interpretation required, predictability of events). As a consequence, any given observation of the performance of a given architecture on a given environmental scenario is likely to reflect the combined effects of many such variables.

Controlled experimentation is an attempt to reduce as much as possible the incidental variability in a set of observations in order to: (a) obtain a reliable account of particular effects of a particular set of theoretically interesting variables; and (b) rigorously bound the class of situations in which those effects can be expected to obtain. In a controlled experiment, one or more "independent variables" are manipulated and their distinctive effects on one or more "dependent" variables are measured. In the present context, we will typically be manipulating independent variables representing a proposed architecture within *S* and

measuring dependent variables representing the performance of interest within  $E$ . Other variables, including both  $S$  and  $E$  variables, are "controlled" to avoid confounding their effects with those of the independent variables. We also will often evaluate the performance of a fixed  $S$  as a function of various  $E$  scenarios; for this, we keep  $S$  fixed and the dependent and independent variables are in  $E$ .

Some controlled variables are simply held constant, while others are systematically manipulated or randomly sampled to provide a basis for generalization. In the domain of IRTPS systems, there are a variety of important variables we may wish to control:

**$S$  variables:**

- sensors
- effectors
- knowledge available
- computing resources
- responsibilities

**$E$  variables:**

- domain
- rate of critical/non-critical events
- distribution of deadlines
- complexity of events (e.g., multi-variate, temporal properties, noisy, uncertain)
- complexity of required effects of actions
- complexity of reasoning required
- knowledge required
- tasks required

In general, the more thoroughly variables are sampled within a class, the more reliably we can generalize conclusions based on associated observations. Statistical inference techniques permit probabilistic statements about the likelihood that relations observed in a given number and distribution of instances will hold for the entire class of such instances.

For example, to compare two different control mechanisms,  $A$  and  $B$ , we might set up two complete agent architectures differing only on this single independent variable, while holding constant all other architectural variables. In effect, we would be placing the  $S - E$  boundary tightly around the control mechanism. We might then apply the architectures to a set of scenarios in which we hold constant the environmental domain (e.g., power plant

monitoring) and systematically manipulate the frequencies of critical and non-critical events and the associated distributions of deadlines. In each condition (unique combination of values of independent and controlled variables), we would measure several dependent variables, such as logical correctness of response and satisfaction of deadlines for critical and non-critical events. Given an appropriate statistical analysis of the results of these measurements, we might draw certain conclusions with high confidence, for example: (a) for critical events in all conditions, control mechanism *A* produces a higher rate of correct answers within deadline than control mechanism *B* (97% vs. 75%); (b) for non-critical events in all conditions, control mechanism *B* produces a higher rate of correct answers within deadline than control mechanism *A* (75% vs. 65%); (c) as the frequency of events increases (1→50 events per unit time), control mechanism *A*'s performance on critical events degrades slowly (100→95%), while its performance on non-critical events declines dramatically 90→40%; and (d) as the frequency of events increases, control mechanism *B*'s performance declines significantly for both critical and non-critical events (95→50% in both cases). Given a particular set of utility functions—in particular, valuing critical events more highly than non-critical events—we might conclude from these results that control mechanism *A* is “better” than control mechanism *B* because it provides “better” performance overall and a “better” degradation profile.

Control of variables determines what kinds of conclusions an experiment can support. For example, observing the performance of  $S_1$  and  $S_2$  on a single scenario,  $O_1$ , in a single environment,  $E_1$ , permits only conclusions about the comparative effectiveness of  $S_1$  and  $S_2$  on scenario  $O_1$ . Observing performance on a representative sample of scenarios in  $E_1$  permits conclusions about the comparative effectiveness of  $S_1$  and  $S_2$  in environment  $E_1$ . Observing performance on a sample of scenarios in a sample of environments from class  $E$  permits conclusions about the comparative effectiveness of  $S_1$  and  $S_2$  in environment class  $E$ .

Of course some variables are quite difficult to control, and these necessarily limit the conclusions that can be drawn from experiments. In particular, it is difficult to separate and control variables that distinguish among the architecture, implementation, and knowledge of a given system  $S$ . Nonetheless, if we wish to draw strong conclusions about the utility of an architecture, we must control these potentially confounding variables. In many cases, our experiments may permit conclusions only about the level of performance of an  $S$  or the relative performances of alternative  $S$ s. It may require analytical methods—or be infeasible—to determine whether an  $S$ 's performance advantage is due to its architecture, implementation, or knowledge.

### 3.5 Other Implications

Although the primary purpose of this document concerns methodologies for evaluating IRTPS architectures, these considerations also carry implications regarding requirements specification and experimental testbed.

### 3.5.1 Implications for Requirements Specification

At this early stage of IRTPS research, we have many proposals for IRTPS requirements that use idiosyncratic, but overlapping vocabularies to describe concepts whose relationships to one another are ambiguous. The proposed model of embedded systems offers an opportunity to operationalize intuitive concepts like reactivity, coherence, interruptability, etc. in terms of basic measurements on state values. Thus, it will become clear, for example, that researcher A uses the term "interruptability" to refer to the latency to respond to a critical event, while researcher B uses the same term to refer to the ability to abort an ongoing computation. Although this does not insure agreement on terms and definitions, it provides a "lingua franca" for communication about terms and definitions.

The concept of an  $S - E$  boundary is fundamental to the proposed model of embedded systems. Although placement of the boundary is flexible, to allow study of IRTPSs of differing scopes, a given placement of the boundary structures the definition of requirements and assessment of their satisfaction. Thus, on one side of the boundary, we have the  $S$  whose "requirements" constitute a theory or design that is hypothesized or intended to produce satisfactory consequences in  $E$ . On the other side of the boundary, we have the  $E$  whose "requirements" define the so-called satisfactory consequences. An informative experiment will tell us to what degree the requirements hypothesized for  $S$  (and realized in a particular implementation) actually achieve the requirements demanded for  $E$ .

For example, we call systems "reactive" in (at least) two different situations. First, we speak of reactive systems that iterate a highly efficient sense-act loop. Second, we speak of systems as reactive if they react promptly to important external events. Under the proposed model of embedded systems, the first sense of reactive is a hypothesized requirement on  $S$ ,  $S$ -Reactivity while the second is a defined requirement on  $E$ , call it  $E$ -Reactivity. The testable claim is that  $S$ -Reactivity produces  $E$ -Reactivity (and perhaps some other desirable  $E$ -Requirements as well). Notice, however, that other testable claims are possible, for example that a non-reactive  $S$  (one whose architecture is something different from the above-mentioned sense-act loop) also produces  $E$ -Reactivity (and perhaps some other desirable  $E$ -Requirements as well). The purpose of experiments is to evaluate such claims.

Accordingly, the model of embedded systems suggests that efforts to specify requirements for IRTPSs clearly distinguish between  $S$ -Requirements and  $E$ -Requirements. In particular,  $E$ -Requirements should be defined strictly in terms of measurements on  $E$  variables, while  $S$ -Requirements should be defined in terms of measurements on  $S$  variables. For example,  $S$ -Reactivity might be defined as a bound on the computation performed between sensing and acting.  $E$ -Reactivity might be defined as a bound on the latency between occurrence of an important "problem" event and the occurrence of an appropriate external "correction" event. Moreover, a given experiment requires agreement among participants on the  $E$ -Requirements against which alternative  $S$ -Requirements will be evaluated.

### 3.6 Implications for an Experimental Testbed

As discussed throughout this document, IRTPSs are complex artifacts embedded in complex environments. Experiments that allow generalization of conclusions beyond the immediate experimental conditions require control of many variables in both the *S* and the *E*. In addition, experimentation on *S*'s of differing scopes requires flexibility in the placement of the *S* -- *E* boundary. To support these kinds of experimentation requires a sophisticated testbed.

For example, a basic testbed would allow one with an *S* to run and experiment with it. The testbed provides an *E* (likely simulated, and also perhaps parameterized) and defines the *S* -- *E* boundary by the interface functions and data structures through which *S* and *E* interact. The testbed should also provide means for controlling multiple runs and for collecting measurements on *E* and, perhaps, on *S* as well. Ideally a testbed also provides utilities for analyzing the results (i.e., the measurements) across multiple runs.

A more general testbed facility would not have the *E* built in, but would instead accept both the *E* and the *S* as inputs. That is, it defines a generic interface between any *E* and any *S* compatible with that *E*. It would accomplish this by defining an interface between the testbed itself and *S*, and between the testbed and *E*. These interfaces would allow the testbed to control each of *S* and *E*, to support their interactions, and to collect the measurements.

For still more flexibility, a testbed would support experiments with varying boundaries between *S* and *E*. That is, the experimenter would supply a complete, modular system to the testbed and specify, for any run, where the *S* -- *E* boundary is. This requires a more extended interface definition facility -- one which supports a complex of interacting modules, preferably composable, and allows for measurements on any of their interfaces. Indeed, we can generalize this concept so that the *S* -- *E* boundary is not even fixed for a run, but shows up only in terms of which measurements are designated the utility measurements. This reflects the idea that the *S* and *E* together are a complete system, and specifying the boundary is only a means for analysis and evaluation.

## Chapter 4

# Report on Research Issues Session of IRTPS Workshop

### 4.1 Introduction

The working group on *research issues* was given the following set of tasks:

1. Propose IRTPS-related research areas.
2. Consider specific research topics relevant to IRTPS within each area.
3. Estimate each topic's potential value to IRTPS and its difficulty.
4. Identify research areas and topics *less* suitable for important, near-term IRTPS research.

Our deliberation was to include non-AI research areas, and we were asked to list research areas in other disciplines (both within and outside of computer science) that are important to IRTPS and suggest the most important IRTPS-related research topics within these non-AI areas.

We began by considering the following four-stage model of how research on a major problem like IRTPS might progress:

1. Paradigms (originally labelled "models")
2. Theoretical questions
3. Algorithm development
4. Technology base (tools and design principles)

Although the discussants generally felt that all of these stages are important to IRTPS, the discussion focused primarily on the first two stages and the relationships between them.

The discussion of this model of scientific progress produced the following:

1. Instead of "paradigms" the term "models" was originally used for the first stage. However, subsequent discussion revealed that we had a more general range of activities in mind than the descriptive activity of modeling. Paradigms entail broad commitments to (a) prioritization of issues to be addressed, (b) a general vocabulary and set of constraints on its use in describing the phenomena of interest, and (c) preferred methodology for assessing alternative descriptions (predictions) and techniques produced within (i.e., consistent with) the paradigm. Therefore, commitment to a paradigm constrains its user both conceptually and methodologically in explorations of some set of phenomena. Initial activity at this stage centers around very informal distinctions and constructs. For quite some time there may be many, equally plausible, candidate paradigms. As a field matures the predominant paradigms are likely to yield formalized theory languages within which detailed models can be specified. Prioritization of research issues is also significantly sharpened, and the methodology for measuring progress is also formalized.
2. The role of "theoretical questions" was less clear. Some participants in our discussion emphasized how paradigms enable the formulation and examination of theoretical questions. In this sense, maturation of a paradigm logically precedes the theoretical examination used to refine that paradigm. Other participants emphasized the many important theoretical questions that must be asked and answered in order to formulate a paradigm in the first place. Such theoretical questions may be "pre-paradigmatic," or even "non-paradigmatic." In this alternative sense, examination of certain theoretical questions logically precedes articulation of a paradigm.
3. Both of these views are valid. They provide complementary accounts of the relationship of theory development to analysis and experimentation. By combining these accounts one comes to see scientific progress as a complex, evolutionary process rather than a simple sequential ordering of stages.
4. The theoretical questions that may be asked at the pre-paradigmatic stage are more likely to focus on broad issues and to be only informally stated. However, these informal questions play a very important role in the early evolution of a scientific discipline by helping to articulate and prioritize the concepts and methodology on which some more formal paradigm will stand.
5. For computationally oriented topics such as IRTPS, algorithm development (stage 3) and technology creation (stage 4) play an essential role as conclusive analytic and empirical tests, respectively, of the concepts and methods produced by a paradigm.

This broad discussion of scientific progress provided us a useful context within which to focus more specifically on the IRTPS initiative:



1. IRTPS research is at an early stage of evolution. Consequently, it is important to support careful, though informal, examination of concepts and issues that may point the way toward choosing among competing paradigms or creating a new, more appropriate paradigm. There are many candidates for the role as a dominant IRTPS paradigm. These need to be better understood and evaluated.
2. IRTPS researchers would benefit greatly from the existence of "paradigm problems," simplified problem instances whose features exemplify critical and broadly occurring features of the full range of IRTPS problems. For example, the well-known Sussman anomaly within the M.I.T. "blocks worlds" serves as a paradigm problem that has stimulated much work within AI on reasoning about action. IRTPS needs its own stock of paradigm problems. However, it is not likely that one may successfully set out to formulate a paradigm problem. A problem achieves this status only if it becomes a widely studied problem, whose interpretation is generally agreed to.

In keeping with our discussion of the evolution of research, we attempted to generate plausible paradigms (models) around which IRTPS research could be organized. The next section presents the results of this analysis. We then exploited our knowledge of these paradigms to generate candidate research areas upon which effort might be focused in the IRTPS initiative, to refine topics within these research areas, and evaluate their significance for IRTPS.

## **4.2 Candidate IRTPS Paradigms**

Challenging concepts underlie the idea of an IRTPS system. Scientists are far from agreeing on the meaning of the concept of an "intelligent system" (whether computational or not). Nor are they ready to commit to a common understanding of what "problem solving" entails. Even the term "real-time" tends to generate vigorous debate. A number of disciplines suggest alternative perspectives and approaches to the development of IRTPS systems. Some of the more prominent candidates include the following:

### **4.2.1 Planning**

This paradigm represents the AI mainstream. Problem-solving is conceived as goal-driven reasoning about action, planning, and "meta-planning." Advocates of this paradigm expect that further research will enable them to extend its basic concepts and techniques so that these methods exhibit real-time performance.

### **4.2.2 Logics**

This paradigm is popular at Stanford's Center for the Study of Language and Information, among other places. It proposes to use special extensions of classical first-order predicate

calculus to formally describe problem-solving, and now IRTPS, in terms of the interactions of an agent's beliefs, desires, and intentions. One set of logic-based approaches involves extending classical logic with modal operators, axioms, and inference rules intended to capture the "logic" of reasoning with beliefs, desires, etc. Another approach involves adding mechanisms for so-called "non-monotonic reasoning" that allow inference with assertions that may under some circumstances be retracted.

### 4.2.3 Algebraic Techniques

This paradigm was suggested by one participant in our discussion as an alternative to logic-based approaches. The same goals hold but, in this paradigm, one uses and extends such formal constructs as universal algebra and similar formal systems. These algebraic techniques, rather than predicate logic, provide the underlying formal system.

### 4.2.4 Decision Theory

This paradigm views problem-solving as choice among alternative actions on the basis of preference. In this view, problem-solving is represented as choosing those actions that maximize a problem-solver's subjective utility. It puts forward the principle of maximizing (subjective) expected utility as a "gold standard" for characterizing ideal rational action. Advocates of this paradigm suggest that real-time problem-solving involves making rational choices while also taking into account the costs of thinking (i.e., computation) and of missing deadlines. IRTPS is thus conceived of as entailing a process to manage scarce problem-solving *resources* such as time and information. Decision theory is sufficiently mature as a paradigm to have well-tested mathematical methods for explicating its view. The formal methods of decision theory enable the explicit modeling of the problem-solver's probabilistic uncertainty, its preferences, and its attitude towards risk.

### 4.2.5 Control Theory

This paradigm has culminated in a mature academic discipline that has produced a large amount of practical technology for building real-time systems. Modern control systems manage complex distributed physical processes, and often do so while meeting real-time performance constraints. Control theory has also produced practical technology for building learning systems. Modern control theorists often conceive of an advanced control system as a non-trivial embodiment of intelligent (or at least intelligently created) problem-solving processes. Control theory and decision theory are closely related. Indeed, the theoretical language of decision theory can be used to describe many of control theory's concepts and formal constructs. However these paradigms differ in some important ways. In particular, applications of control-theoretic methods often model a "closed loop" relationship between the actions of a controlling system and the controlled system that it manages. By contrast, the formal methods of decision theory make it difficult to model feedback. Secondly, advanced control theories have been developed that emphasize specifying a control response

by analyzing the feedback and other observations of the "plant" (the controlled system) in terms of a reference model available to the controlling system. Model-reference methods can emphasize qualitative issues that have received less attention in decision theory. Examples include use of a reference model that only approximately models the actual dynamics of the plant and methods for replacing a state-based model with one that has fewer states but which can still be used to meet the original control criterion (i.e., so-called state-aggregation methods). In sum, although control theory and decision theory are closely related paradigms, they provide significantly different emphases and may even turn out to be formally incompatible.

#### **4.2.6 State-Based Automata Methods**

One discussant proposed state-based automata methods as an alternative paradigm. The principal distinction of this paradigm seems to lie in its use of the mathematics of automata theory for formal modeling and analysis. There may also be some increased emphasis upon the control of non-linear systems in this approach, although this difference from standard control theory has diminished in recent years. In fact, many state-based methods are well-known to control theorists, and many of the standard problems defined within the state-based paradigm are similar or identical to the standard problems emphasized by control theorists, e.g., reachability, identifiability, stability, convergence, and controllability.

#### **4.2.7 No-Paradigm**

Several of the participants pointed out that IRTPS (and AI in general) is arguably at a very early, pre-paradigmatic stage. No existing paradigm obviously provides explicit coverage of all IRTPS issues. In particular, no existing approach can readily claim to have been formulated explicitly with IRTPS in mind. So, we should not attach ourselves to one of these existing paradigms with the illusion that it provides a suitable basis for investigating IRTPS. Given that this is the case, there are at least three alternative courses of action:

1. Do something to develop a new paradigm from scratch. Unfortunately, it is difficult to imagine how to guide this process, or even to suggest how work of this type could be evaluated or supported. Most likely, one can judge the value of a new paradigm only in retrospect.
2. Compare alternative existing paradigms and then adopt the best ideas from the closest or some combination. This variant probably describes the approach of many AI researchers in the history of that field. The paradigm labelled "Logics" above may exemplify this to a certain extent.
3. Look for some deeper, more foundational paradigm within which IRTPS is a "special case." This alternative is strongly reductionist and presumes that one has a coherent sense of the dimension of reduction and a convincing argument for the acceptability of the chosen paradigm for the underlying level. As an example, some AI researchers have attempted to rely upon work at the foundation of mathematics as a deeper conceptual

and formal basis for their approach to AI problems. Prima facie, this strategy may be problematic for IRTPS because of foundational difficulties in modeling notions of *process* and *time*.

## 4.3 Toward an IRTPS Research Agenda

Our discussion group contained advocates of all the candidate paradigms discussed above. We exploited these diverse perspectives in generating a set of candidate IRTPS research areas.

### 4.3.1 Research Areas

After some discussion and re-organization, our group's participants agreed on a rough outline of the areas and sub-areas within which they expect to find research problems that are critical to IRTPS. They distinguished these areas as being either issues of resource-constrained reasoning or problems of modeling a real-time environment. The subsequent discussion of topics (reviewed in the next section) focused only on the former (with some rearrangement.)

The following outlines critical IRTPS-related research areas:

1. Problem-solving under resource constraints
  - (a) Limited resource reasoning
    - i. Controlling focus of attention
    - ii. Hierarchy of reflection
    - iii. Temporal reasoning
    - iv. "Anytime" reasoning
  - (b) Managing varying resources at "runtime"
  - (c) Managing competing objectives
  - (d) Reasoning about resources
2. Modeling features of a complex, dynamic environment
  - (a) Uncertainty and unpredictability
  - (b) Time-dependent events and action outcomes

### 4.3.2 Prioritization

After settling on this preliminary categorization of IRTPS research issues, our afternoon session focused on two subsidiary goals: First, we attempted to identify fairly specific research questions within each of the broad areas that had been discussed in the morning and second, we attempted to evaluate each of these research questions with regard to three properties:

1. How likely was it that progress would be made in this specific area in the next two years?
2. How important to the IRTPS venture would this progress be?
3. How "easy" was the problem? This term was not defined further, but seemed to measure the group's collective confidence that progress predicted in (1) would in fact be achieved.

Of course, all of the decisions made about these questions were entirely subjective, and should be taken principally as reflecting the opinions and biases of the research issues working group.

### **Short-Term, Important Issues**

The following research problems were felt to be important ones on which progress could be expected in the near term:

**Compilation** The problems here are those of preprocessing information so that it can be accessed more quickly when it is needed, or transforming one representation of a process to another, more efficient, representation. The techniques mentioned included inductive methods, the synthesis of causal theories, case-based reasoning, and neural networks. This is such an active research area already that it was felt certain that progress would indeed be made over the next two years, and compilation was therefore labelled "easy."

**Resource Representation and Monitoring** What problems are involved in representing the fact that IRTPS systems can be expected to encounter resource limitations? What needs to be done to make it practical for these systems to monitor their resource consumption and needs? The first of these was felt to be easy; the second, less so.

**Time-Dependent Utility** How can an agent expect the utility of achieving certain goals to vary over time? Not easy.

**Metareasoning Applied to Planning and to Search** Applying some sort of metalevel reasoning to control search, to determine when replanning is needed, or to weigh competing subgoals. Easy.

**Sensory Planning** Planning to acquire new information through the use of sensors. Easy.

**Plan Monitoring** Given that a plan has been constructed to achieve some real-time goal, how can progress be monitored as the plan is executed? Not easy.

**Anytime Reasoning** Here, we identified a variety of subproblems that could be lumped under the title of "anytime inference:"

1. Redefining inference in a way that allows the inference process to be interrupted, and constructing algorithms that capture this new definition.
2. Constructing measures on the value of an incomplete answer to a declarative query.
3. Defining a notion of an "approximate" answer to a declarative query. (Not short-term.)
4. Anytime planning, by which we mean the development of planning systems that work by progressively improving/refining partial plans.

None of these problems was felt to be easy.

**The Application of Decision Theory to the Problem of Focusing Attention** What should the agent concentrate on next? Which of its sensors should be polled at any particular time? Not easy.

Finally, we included in this group of problems that of investigating the foundations of intelligent, real-time problem solving. It was felt that progress would inevitably be made here.

### **Important Issues, But Not Short-Term**

These problems were generally felt to be harder than those in the previous section:

1. Resource planning. Planning to obtain more resources.
2. Process representation. Understanding and representing the time-dependent processes with which IRTPS systems will need to interact.
3. Utility of metareasoning and learning. Given that we reason about our own problem-solving activity and that we learn, under what circumstances are these mental activities useful ones?
4. Understanding and being able to deal with competing goals.
5. Understanding dynamic focus of attention. How is it that an agent's focus of attention changes from time to time? What should control this process?
6. Understanding partial or incomplete plans; debugging incorrect plans.

## Not Important

Finally, several research areas were identified that were felt to be of lesser importance to the IRTPS effort. In some cases, this was because these problems were not felt to lie within the IRTPS domain specifically; in others, the problems simply failed to generate any enthusiasm within the research issues working group.

1. Learning metalevel information by induction.
2. Caching. By this we mean the study of data structures and mechanisms that could be used to save previously computed results.
3. Metalevel reasoning applied to scheduling, to situation understanding, and to probabilistic computation.
4. The application of operating systems concepts (interrupts, priorities, etc.) to the problem of focusing attention.
5. Defining the concept of a plan.
6. Planning for goals that are partially satisfiable or temporally scoped (such as that of maintaining some external condition). This and the previous topic were felt to be the domain of the planning community proper and outside the scope of IRTPS per se.

## 4.4 Concluding Remarks

The working group was aware that the background and interests of its members may not have been fully representative of the IRTPS community as a whole and may have introduced certain biases into its conclusions. For example, the lack of representation of more researchers with strong applications experience probably limited our ability to generate a comprehensive list of the "right" questions. We were also restricted in our ability to propose paradigms because important, non-AI disciplines were under-represented. The "AI planning" contingent, on the other hand, was proportionately over-represented, and as a result, our discussion of research topics tended to be biased towards issues within that paradigm. These observations about the composition of the group are offered not to diminish the validity of our conclusions, but rather to place them in their appropriate context.

## **Chapter 5**

### **Annotated Bibliography**

The following is a bibliography containing a representative collection of references to literature relevant to intelligent real-time problem-solving. Each item has been annotated with a brief description of its content and its relation to the field.



# Bibliography

- [1] J.A. Barnett. How much is control knowledge worth? *Journal of Artificial Intelligence*, 22(1):77-89, January 1984. A relatively simple, but powerful example of the value of applying metareasoning to raise the probability that an appropriate order of strategies will be applied to a problem instance.
- [2] M. Boddy and T. Dean. Solving time-dependent planning problems. In *Proceedings of the Eleventh IJCAI*. AAAI/International Joint Conferences on Artificial Intelligence, August 1989. A discussion of limitations in classic AI planning techniques and of the application of flexible reasoning strategies, called "anytime algorithms," to the problem of planning under limitations in resources. Article also includes a description about a research direction focused on the application of decision theory to control reasoning.
- [3] R. Brooks. Planning Is Just a Way of Avoiding Figuring Out What to Do Next. Technical Report Working Paper 303, M.I.T. Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1987. A discussion of the fundamental goals of planning. Discussion directs research beyond complex deliberative machinery, and towards more tractable reactive reasoning systems.
- [4] R.M. Chavez and G.F. Cooper. A fully-polynomial randomized approximation scheme for the bayesian inferencing problem. *Knowledge Systems Laboratory Technical Report 88-72*, October 1988. A discussion of work on the analysis of a simulation approach to performing the probabilistic inference problem. The method gives a worse-case analysis of inference as a function of a variety of parameters from the probability model.
- [5] G.F. Cooper, E.J. Horvitz, and D.E. Heckerman. A model for temporal probabilistic reasoning. Technical Report KSL-88-30, Stanford University, Knowledge Systems Laboratory, Stanford, CA, April 1988. A discussion of the problem of temporal reasoning under uncertainty. First the general problem of uncertain reasoning through time is introduced. After, assumptions that reduce the complexity of the analysis are presented. The notion of general and stereotypical temporal functions are introduced. Following the exposition of a theory for combining uncertain beliefs over time, the paper presents work on an implementation of the techniques. System output is included in the discussion.
- [6] B. D'Ambrosio. Process, structure, and modularity in reasoning with uncertainty. In *Proceedings of Fourth Workshop on Uncertainty in Artificial Intelligence*, Minneapolis, MN, July 1988. American Association for Artificial Intelligence. A discussion of the

importance of considering details of the structure and use of the probabilistic model in reasoning under uncertainty.

- [7] T. Dean. On the value of goals. In J. Tenenbergs, J. Weber, and J. Allen, editors, *Proceedings from the Rochester Planning Workshop: From Formal Systems to Practical Systems*, pages 129-140, Rochester, 1989. American Association for Artificial Intelligence. An interesting discussion of the application of decision theory to planning. Discussion examines the assignment of utility to goals in planning systems.
- [8] J. Doyle. Artificial intelligence and rational self-government. Technical Report CS-88-124, Carnegie-Mellon University, 1988. A discussion of a distributed approach to rationality based on the fair, democratic interaction among independent processes. Paper describes a language for reasoning about important predicates and functions in a distributed rational system.
- [9] M.R. Fehling and J.S. Breese. A computational model for the decision-theoretic control of problem solving under uncertainty. Technical Report 837-88-5, Rockwell International Science Center, April 1988. An example analysis of the decision-theoretic control of robot planning that considers base-level as well as meta-level considerations. The article considers issues of decision making under limitations in knowledge and computational time.
- [10] O. Hansson and A. Mayer. The optimality of satisficing solutions. In *Proceedings of Fourth Workshop on Uncertainty in Artificial Intelligence*, Minneapolis, MN, July 1988. American Association for Artificial Intelligence. An early discussion of the control of search with decision theory. The discussion highlights tradeoffs in the problem of search. The 8-puzzle is used as an example to elucidate the possibility of controlling search with decision theory. Empirical results of work on different dimensions of search are included.
- [11] D.E. Heckerman, J.S. Breese, and E.J. Horvitz. The compilation of decision models. In *Proceedings of Fifth Workshop on Uncertainty in Artificial Intelligence*, Windsor, Canada, August 1989. American Association for Artificial Intelligence. A discussion of the optimal compilation of a decision problem, given (1) the stakes of a decision problem, (2) the cost incurred with computational delay, (3) a characterization of possible observations by a distribution over evidence weights, and (4) the cost of memory. The foundations of decision-theoretic compilation are introduced. Design-time selection rules for configuring optimal sets of rules are described. Analyses for different prototypical distributions of evidence weights are presented. Finally, the paper discusses issues surrounding the relaxation of assumptions, and highlights the usefulness of developing mixed compiled-compute strategies.
- [12] D.E. Heckerman and E.J. Horvitz. On the expressiveness of rule-based systems for reasoning under uncertainty. In *Proceedings AAAI-87 Sixth National Conference on Artificial Intelligence*, pages 121-126, Seattle, Washington, July 1987. American Association for Artificial Intelligence. An analysis of pragmatic issues surrounding the efficiency of the production-rule versus the belief-network representation approaches for the problem of performing diagnosis under the general situation of uncertainty. The

production-rule representation is shown to be limited in its ability to express efficiently important independencies and patterns of independence among distinctions in a knowledge base.

- [13] D.E. Heckerman, E.J. Horvitz, and B.N. Nathwani. Toward normative expert systems: The Pathfinder project. *Computers and Biomedical Research*, 1989. In press. A detailed review of the Pathfinder project to construct large probabilistic knowledge bases that contain generalized dependencies, and to perform inference tractably within these large models. A highlight of this work centers on the use of probabilistic similarity networks, a theoretically sound representation for focusing the attention of knowledge representation.
- [14] E.J. Horvitz. Reasoning about inference tradeoffs in a world of bounded resources. Technical Report KSL-86-55, Stanford University, Knowledge Systems Laboratory, Stanford, CA, September 1986. An early description of the use of multiattribute decision theory to control partial-computation strategies. The extensions of decision theory to include metareasoning about base-level inference are highlighted. A set of bounded-resource desiderata are enumerated for strategies that are used to solve problems under situations of varying or uncertain resource constraints.
- [15] E.J. Horvitz. Towards a science of expert systems. In Thomas J. Boardman, editor, *Proceedings of the 18th Symposium on the Interface of Computer Science and Statistics*, pages 45-52, Ft. Collins, Colorado, March 1986. American Statistical Association. Also available as Technical Report KSL-86-75, Knowledge Systems Laboratory, Stanford University, March 1986. Examines the promise of reasoning about computational and engineering tradeoffs to optimize the value of reasoning systems. The paper highlights the value of approximations for representations and computation that exhibit graceful degradation of performance, with decreasing allocation of engineering or reasoning resource. Formal research on the graceful degradation of value-of-information computation for use in the Pathfinder expert system is presented. This approach is based on reformulating information theory to handle groups of diseases, versus single diseases, under consideration in a medical expert system.
- [16] E.J. Horvitz. Problem-solving design: Reasoning about computational value, tradeoffs, and resources. In *Proceedings of the NASA Artificial Intelligence Forum*, pages 26-43. National Aeronautics and Space Administration, November 1987. Also available as Technical Report KSL-87-64, Knowledge Systems Laboratory, Stanford University, October 1987. Discussion of the decision-theoretic design of problem solving versus real-time decision-theoretic control. Control reasoning is divided up into "strategic" and "structural" control; control based on determining a best order to apply well-defined computational policies or "strategies," is distinguished from control centering on the manipulation at the microstructure of an algorithm, to optimize its performance in some context. Finally, the paper discusses the feasibility of developing a formal science of limited rationality, by developing tools for design and control of computation within the framework of decision theory. Problems in reasoning under complexity and constraints of medicine are described.

- [17] E.J. Horvitz. Reasoning about beliefs and actions under computational resource constraints. In *Proceedings of Third Workshop on Uncertainty in Artificial Intelligence*, Seattle, Washington, July 1987. American Association for Artificial Intelligence. Also in L. Kanal, T. Levitt, and J. Lemmer, ed., *Uncertainty in Artificial Intelligence 3*, Elsevier, pps. 301-324. Early discussion of problems with traditional conception of rationality from decision science, as well as with heuristic solutions in AI research. Both naive decision-theoretic approaches and heuristic, satisficing approaches are viewed as suboptimal and costly. The paper introduces the decision-theoretic control of problem solving as a general framework for attacking the problem of ideal bounded rationality. Several problems, such as ideal compilation and metareasoning tractability are discussed; relevant literature from the decision sciences is highlighted. Concept of "bounded optimality," the optimization of the expected value of a reasoner, under constraints in its reasoning resources and constitution, is presented. Finally, properties desired of partial-computation strategies for reasoning under bounded resources are outlined. Properties of flexible computation include continuity, monotonicity, and convergence. These properties of flexibility are implicit in strategies that later came to be called "anytime algorithms."
- [18] E.J. Horvitz. Reasoning under varying and uncertain resource constraints. In *Proceedings AAAI-88 Seventh National Conference on Artificial Intelligence*, pages 111-116. American Association for Artificial Intelligence, August 1988. Formalization of partial computation under bounded resources. The paper introduces a multiattribute-utility approach to partial results for the control of fundamental computer-science problems, such as searching and sorting, and describes computational trajectories through multidimensional approximation spaces. A mathematical exposition highlights the value of strategies that exhibit monotonic refinement with computation, for reasoning under varying and uncertain resource constraints. Includes an expected-utility analysis of flexible reasoning strategies under different classes of resource constraints. Resource classes include the situations of urgency, deadline, uncertain deadline, and urgent-deadline (a combination of the urgency and deadline situations). Beyond theoretical work, the paper describes empirical results from the performance of the Protos/Algo system, on the decision-theoretic control of fundamental computation tasks.
- [19] E.J. Horvitz. Optimal allocation of resources to metareasoning and control. Technical Report KSL-89-66, Stanford University, Knowledge Systems Laboratory, Stanford, CA, August 1989. A discussion of the optimal allocation of resources to metalevel deliberation or planning, versus to execution. Several types of metareasoning and control are reviewed, and prototypical cases are analyzed.
- [20] E.J. Horvitz. Rational metareasoning and compilation for optimizing decisions under bounded resources. In *Proceedings of Computational Intelligence 89*. Association for Computing Machinery, September 1989. A discussion of the relationship between decision-theoretic control of reasoning and the compilation of reasoning. The paper reviews the construction of integrated reasoning systems that allow for the compilation of portions of deliberative processes at the base-level and meta-levels. Assumes a basic metareasoning architecture and allows for the introduction of several classes of com-

piled knowledge. Finally, the paper presents a view of learning as an agent's attempt to optimize its reasoning in a specialized environment, through local and long-term compilation.

- [21] E.J. Horvitz. Some fundamental problems and opportunities from the standpoint of rational agency. Technical Report KSL-89-28, Stanford University, March 1989. A review of some outstanding problems and research opportunities in the pursuit of "bounded-optimal" agents under resource constraints, presented at the AAAI Spring Symposium on Limited Rationality at Stanford, March, 1989. Issues discussed include problems at the foundations of preference regarding the optimization of long-term versus myopic utility, difficulties with evaluating the relative performance of different agents, problems associated with analytic regress, difficulties with the ideal integration of compiled and deliberative machinery, and problems with the analysis and validation of optimal partial-computation strategies.
- [22] E.J. Horvitz, G.F. Cooper, and D.E. Heckerman. Reflection and action under scarce resources: Theoretical principles and empirical study. In *Proceedings of the Eleventh IJCAI*, pages 1121-1127. AAAI/International Joint Conferences on Artificial Intelligence, August 1989. Introduces a formal model of rationality, based on the control of complex base-level inference by relatively tractable decision-theoretic metareasoning. The paper addresses the question, "How long should an agent deliberate about a problem before acting in the world?" The authors show that the answer to this question depends on (1) the stakes of the situation at hand, (2) the costs of deliberation, and (3) metaknowledge about the expected value of continuing to reason. In the general case, there may be uncertainty about all of these factors. Theoretical principles of belief and action under bounded resources are presented, and results of experiments with a complex decision problem, under resource constraints, are described.
- [23] E.J. Horvitz, D.E. Heckerman, and C.P. Langlotz. A framework for comparing alternative formalisms for plausible reasoning. In *Proceedings AAAI-86 Fifth National Conference on Artificial Intelligence*, pages 210-214, Philadelphia, Pennsylvania, August 1986. American Association for Artificial Intelligence. Examines the promise of reasoning about computational and engineering tradeoffs to optimize the value of reasoning systems. The paper highlights the value of approximations for representations and computation that exhibit graceful degradation of performance, with decreasing allocation of engineering or reasoning resources. Formal research on the graceful degradation of value-of-information computation for use in the Pathfinder expert system is presented. This approach is based on reformulating information theory to handle group of diseases, versus single diseases, under consideration in a medical expert system.
- [24] E.J. Horvitz, D.E. Heckerman, K.C. Ng, and B.N. Nathwani. Heuristic abstraction in the decision-theoretic Pathfinder system. In *Proceedings of the 13th Symposium on Computer Applications in Medical Care*, pages 178-182, October 1989. Describes the graceful control of abstraction as a class of partial computation. Abstraction of decision model is used to gain clarity and greater tractability. A difficult decision problem is decomposed into a hierarchy of goals at successively greater detail. In particular, the paper

reviews the complexity of complete value-of-information analyses in the Pathfinder reasoning system, and show how the the evidence-gathering formalism can be applied to abstractions. The use of multiple abstraction hierarchies to control the focus of attention of a decision-theoretic analysis is introduced. Alternative hierarchies allow a problem to be probed and solved from different perspectives. Details on the implementation and interface associated with abstraction modulation in the Pathfinder system are presented.

- [25] E.J. Horvitz, H.J. Suermondt, and G.F. Cooper. Bounded conditioning: Flexible inference for decisions under scarce resources. In *Proceedings of Fifth Workshop on Uncertainty in Artificial Intelligence*, Windsor, Canada, August 1989. American Association for Artificial Intelligence. A description of a new flexible (or "anytime") algorithm for probabilistic inference called "bounded conditioning." The general probabilistic-inference problem is NP-Hard. Bounded conditioning provides upper and lower bounds on a probability of interest; the bounds are tightened incrementally with computation, converging on a point probability with sufficient computation. The method is based on the decomposition of a difficult problem into a set of subproblems, by conditioning the problem on the truth of a spectrum of plausible contexts. The subproblems are solved in an order that guarantees that the most important aspects of the problem will be addressed first. The paper contains an analysis of why a great portion of the complete exponential problem is solved when only a fraction of the complete problem has been analyzed. In addition to the theoretical analyses, empirical study of the performance of the algorithm are described.
- [26] Lesle P. Kaelbling. Rex: A symbolic language for the design & parallel implementation of embedded systems. In *Proceedings of the AIAA Conference of Computers in Aerospace*, 1987. A language for implementing embedded systems should facilitate the development of real-time programs, have clear formal semantics, allow both numeric and symbolic programming, and be easy to implement on parallel hardware. Rex, a Lisp-based language that has been used to implement a variety of mobile-robot control programs, satisfies these criteria and is a viable alternative for the implementation of embedded systems. Because Rex uses sequential circuits as its model of computation, it is easily adapted to run on parallel hardware, and may also be used to design custom chips.
- [27] Leslie P. Kaelbling and Stanley J. Rosenschein. *New Architectures for Autonomous Agents (tentative title)*, chapter Action and Planning in Embedded Agents. Elsevier Science Publishers, to appear. Embedded agents are computer systems that sense and act on their environments, monitoring complex dynamic conditions and affecting the environment in goal-directed ways. Systems of this kind are extremely difficult to design and build, and without clear conceptual models and powerful programming tools, the complexities of the real world can quickly become overwhelming. In certain special cases, designs can be based on well-understood mathematical paradigms such as classical control theory. More typically, however, tractable models of this type are not available and alternative approaches must be used. One such alternative is the situated-automata framework, which models the relationship between embedded control systems and the external world in qualitative terms and provides a family of programming abstractions

to aid the designer. This paper briefly reviews the situated automata approach and then explores in greater detail one aspect of the approach, namely the design of the action-generating component of embedded agents.

- [28] Leslie Pack Kaelbling. *Reasoning About Actions and Plans*, chapter An Architecture for Intelligent Reactive Systems, pages 395–410. Morgan Kaufmann, 1987. Any intelligent agent that operates in a moderately complex or unpredictable environment must be reactive – that is, it must respond dynamically to changes in its environment. A robot that blindly follows a program or plan without verifying that its operations are having their intended effects is not reactive. For simple tasks in carefully engineered domains, non-reactive behavior is acceptable; for more intelligent agents in unconstrained domains, it is not. This paper presents the outline of an architecture for intelligent reactive systems. Much of the discussion will relate to the problem of designing an autonomous mobile robot, but the ideas are independent of the particular system. The architecture is motivated by the desires for modularity, awareness, and robustness.
- [29] Stanley J. Rosenschein. Formal theories of knowledge in AI and robotics. *New Generation Computing*, 3(4, special issue on Knowledge Representation), 1985. Although the concept of *knowledge* plays a central role in artificial intelligence, the theoretical foundations of knowledge representation currently rest on a very limited conception of what it means for a machine to know a proposition. In the current view, the machine is regarded as knowing a fact if its state either explicitly encodes the fact as a sentence of an interpreted formal language or if such a sentence can be derived from other encoded sentences according to the rules of an appropriate logical system. We contrast this conception, the interpreted-symbolic-structure approach, with another, the situated-automata approach, which seeks to analyze knowledge in terms of relations between the state of a machine and the state of its environment over time using logic as a metalanguage in which the analysis is carried out.
- [30] Stanley J. Rosenschein. Synthesizing information-tracking automata from environment descriptions. In *Proceedings of Conference on Principles of Knowledge Representation and Reasoning*, 1989. This paper explores the synthesis of finite automata that dynamically track conditions in their environment. We propose an approach in which a description of the automaton is derived automatically from a high-level declarative specification of the automaton's environment and the conditions to be tracked. The output of the synthesis process is the description of a sequential circuit that at each clock cycle updates the automaton's internal state in constant time, preserving as an invariant the correspondence between the state of the machine and conditions in the environment. The proposed approach allows much of the expressive power of declarative programming to be retained while insuring the reactivity of the run-time system.
- [31] Stanley J. Rosenschein and Leslie Pack Kaelbling. The synthesis of digital machines with provable epistemic properties. In *Proceedings of the Conference on the Theoretical Aspects of Reasoning About Knowledge*, pages 83–98. AAAI, 1986. Researchers using epistemic logic as a formal framework for studying knowledge properties of artificial-intelligence (AI) systems often interpret the knowledge formula  $K(x, \varphi)$  to mean that

machine  $x$  encodes  $\varphi$  in its state as a syntactic formula or can derive it inferentially. If  $K(x, \varphi)$  is defined instead in terms of the correlation between the state of the machine and that of its environment, the formal properties of modal system S5 can be satisfied without having to store representations of formulas as data structures. In this paper, we apply the correlational definition of knowledge to machines with composite structure and describe the semantics of knowledge representations in terms of correlation-based denotation functions. In particular, we describe how epistemic properties of synchronous digital machines can be analyzed, starting at the level of gates and delays, by modeling the machine's components as agents in a multiagent system and reasoning about the flow of information among them. We also introduce Rex, a language for computing machine descriptions recursively, and explain how it can be used to construct machines with provable informational properties.

- [32] S.J. Russell and E.H. Wefald. Principles of metareasoning. In Ronald J. Brachman, Hector J. Levesque, and Raymond Reiter, editors, *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, May 1989. Morgan Kaufman. A review of earlier and ongoing work on the decision-theoretic control of problem solving in AI research. The article discusses several problems that pose challenges to the decision-theoretic control of game-playing search, the focus of the authors' research.
- [33] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1947. The classic discussion of decision theory that presents that axioms of utility and describes a set of issues surrounding the application of probability and utility.
- [34] E. Wefald and S. Russell. Estimating the value of computation: The case of real-time search. In *Working Notes: Spring Symposium Series on AI and Limited Rationality*, pages 106–110, Stanford University, March 1989. American Association for Artificial Intelligence. A discussion of techniques for estimating the value of computation for single-agent search.
- [35] M.P. Wellman. Formulation of tradeoffs in planning under uncertainty. Technical Report MIT/LCS/TR-427, Laboratory for Computer Science, Massachusetts Institute of Technology, August 1988. A detailed discussion of techniques for constructing simple decision models that can dictate optimal action under uncertainty, based on an analysis of fundamental tradeoffs.



# REPORT OF INVENTIONS AND SUBCONTRACTS

(Pursuant to "Patent Rights" Contract Clause) (See Instructions on Reverse Side.)

Form Approved  
OMB No. 0704-0140  
Expire Sep 30, 1988

1a. NAME OF CONTRACTOR / SUBCONTRACTOR TELEOS RESEARCH		c. CONTRACT NUMBER SAME		2a. NAME OF GOVERNMENT PRIME CONTRACTOR AFOSR		c. CONTRACT NUMBER F49620-89-C-0117		3. TYPE OF REPORT (X one) a. INTERIM <input checked="" type="checkbox"/> b. FINAL <input type="checkbox"/>	
b. ADDRESS (Include ZIP Code) 576 MIDDLEFIELD ROAD PALO ALTO, CA 94301		d. AWARD DATE (YYMMDD) 890828		b. ADDRESS (Include ZIP Code) BUILDING 410, BOLLING AFB WASHINGTON, D.C. 20332-6448		d. AWARD DATE (YYMMDD) 890828		4. REPORTING PERIOD (YYMMDD) a. FROM 890828 b. TO 891127	

## SECTION I - SUBJECT INVENTIONS

5. "SUBJECT INVENTIONS" REQUIRED TO BE REPORTED BY CONTRACTOR / SUBCONTRACTOR (If "None," so state)

a. NAME(S) OF INVENTOR(S) (Last, First, MI)	b. TITLE OF INVENTION(S)	c. DISCLOSURE NO., PATENT APPLICATION SERIAL NO. OR PATENT NO	d. ELECTION TO FILE PATENT APPLICATIONS				e. CONFIRMATORY INSTRUMENT OR ASSIGNMENT FORWARDED TO CONTRACTING OFFICER	
			(1) United States		(2) Foreign			
			(a) Yes	(b) No	(a) Yes	(b) No		
NONE	NONE							

g. ELECTED FOREIGN COUNTRIES IN WHICH A PATENT APPLICATION WILL BE FILED

f. EMPLOYER OF INVENTOR(S) NOT EMPLOYED BY CONTRACTOR / SUBCONTRACTOR		g. ELECTED FOREIGN COUNTRIES IN WHICH A PATENT APPLICATION WILL BE FILED	
(1) (a) Name of Inventor (Last, First, MI) NONE	(2) (a) Name of Inventor (Last, First, MI) NONE	(1) Title of Invention NONE	(2) Foreign Countries of Patent Application
(b) Name of Employer	(b) Name of Employer		
(c) Address of Employer (Include ZIP Code)	(c) Address of Employer (Include ZIP Code)		

## SECTION II - SUBCONTRACTS (Containing a "Patent Rights" clause)

6. SUBCONTRACTS AWARDED BY CONTRACTOR / SUBCONTRACTOR (If "None," so state)			
a. NAME OF SUBCONTRACTOR(S)	b. ADDRESS (Include ZIP Code)	c. SUBCONTRACT NO.(S)	d. DFAR "PATENT RIGHTS" (1) Clause Number (2) Date (YYMM)
THE BOARD OF TRUSTEES OF THE LELAND STANFORD JUNIOR UNIVERSITY	SPONSORED PROJECTS OFF JORDAN QUAD/BIRCH 125 PANAMA STREET STANFORD UNIV. STANFORD CALIFORNIA 94305-4125	AGREEMENT NO: 7412	NONE
			SEE ATTACHED SHEET FOR CONTINUATION...
e. DESCRIPTION OF WORK TO BE PERFORMED UNDER SUBCONTRACT(S)			
ASSIST IN RESEARCH ON IRTPS			
			f. SUBCONTRACT DATES (YYMMDD): (1) Award 890915 (2) Estimated Completion 891215

## SECTION III - CERTIFICATION

7. CERTIFICATION OF REPORT BY CONTRACTOR / SUBCONTRACTOR		(Not required if <input checked="" type="checkbox"/> Small business or Non-Profit organization) (X appropriate box)	
a. NAME OF AUTHORIZED CONTRACTOR / SUBCONTRACTOR OFFICIAL (Last, First, MI)		c. I certify that the reporting party has procedures for prompt identification and timely disclosure of "Subject Inventions," that such procedures have been followed and that all "Subject Inventions" have been reported.	
b. TITLE	d. SIGNATURE	e. DATE SIGNED	

# REPORT OF INVENTIONS AND SUBCONTRACTS - CONTINUED

## SECTION II - SUBCONTRACTS

Name of Subcontractor(s)	Address	Subcontract No(s)	Patent Rights	Description of Work to be Performed Under Subcontract(s)	Award	Dates Completion
B. D. D'Ambrosio	Dept. of Computer Science Comp. Sci. Bldg 100 Oregon State Univ. Corvallis Oregon, 97331-3902	---	None	Consultancy	890828	900126
M. L. Ginsberg	Dept. of Computer Science Stanford University Stanford California 94305-2140	---	None	Consultancy	890828	900126
E. J. Horvitz	Knowledge Systems Lab. Depts. of Computer Science and Medicine Medical School Office Bldg. Stanford University Stanford California 94305-5479	---	None	Consultancy	890828	900126